07

Métodos adicionales

Transcripción

[00:00] Hola. ¿Cómo están? Vamos a iniciar nuestra clase 3. En nuestra clase 3 vamos a ver el uso de la clase Collections para ordenar una lista simple. Además también vamos a ver un método sort de la clase ArrayList también para ordenar. Primero, vamos a duplicar nuestra clase. Vamos a colocar clase 3, vamos a remover todo lo que está comentado y en lugar de variable 1, 2, 3 y 4, vamos a colocar ahora curso1, curso2, curso3 y curso4. Vamos a crear una lista de cursos.

[00:45] Vamos a colocar aquí geometría, física, química, historia. En lugar de listaString vamos a colocar cursos, que sería una lista de cursos. Y vamos a colocar aquí, adicionar el curso1, adicionar curso2, adicionar curso3 y adicionar curso4, y vamos a imprimir esta lista. Una vez hecho esto, ejecutamos. Tenemos aquí una lista común adicionada en orden.

[01:34] Entonces se imprimió geometría, física, química, historia. Ahora imaginémonos que queremos ordenar esta lista de forma de menor a mayor, de forma ascendente. Entonces, vamos a utilizar ahora la clase Collection que viene del paquete java.util. Vamos a colocar el método sort y vamos a colocar cursos.

[02:08] ¿Esto qué va a hacer? Va a modificar nuestra lista y va a crear una nueva lista ascendente. Imprimimos nuestra variable cursos y ejecutamos nuevamente. ¿Qué sucedió? Ordenó nuestra lista de menor a mayor, que sería física, geometría, historia y química, en orden de menor a mayor.

[02:39] Luego, ahora imaginémonos que queremos ordenar de forma descendente. Para esto, nuestro método sort también viene con un atributo más, que podemos adicionar, que sería Collections.reverseOrder(). ¿Qué vamos a hacer aquí? Vamos a ordenar de forma descendente. System.out.println, digitamos cursos. Tenemos como resultado nuestra lista de menor a mayor. No, nuestra lista de mayor a menor, nuestra lista de menor a mayor y nuestra lista normal como fue adicionada.

[03:32] Esto sería usando la clase collections. Ahora, supongamos que no quiero utilizar la clase collections, quiero utilizar algún método dentro de la clase ArrayList que nosotros estamos trabajando. Entonces vamos a hacer lo siguiente, vamos a colocar aquí cursos.sort y vamos a utilizar la clase comparator, que sería reverseOrder por ejemplo. Vamos a ordenarlo de menor a mayor.

[04:14] Imprimimos, System.out.println, y colocamos cursos. Ahora vamos a ejecutar. Y tenemos química, historia, geometría y física. Ordenados de forma de mayor a menor. Ahora, queremos de menor a mayor. Utilizamos comparator y el método naturalOrder. ¿Qué va a hacer? Prácticamente va a modificar el orden en nuestra lista de menor a mayor.

[05:00] Ahora, a partir de Java 8, tenemos también el uso de streams. Entonces, también nuestra lista de cursos puede ser también un stream, convertimos en stream y utilizamos el método sorted. Ese método sorted, aquí como ven, también me devuelve un stream. Aquí, una vez hecho esto necesitamos convertir para una lista.

[05:35] Entonces utilizamos el método collect. Para eso usamos la clase Collectors. toList. ¿Qué va a hacer? Me va a devolver una nueva lista de la interface list. Si nosotros ejecutamos ahora, ¿qué va a suceder? ¿Va a ordenar o no? No ordenó. Hizo el mismo orden aquí que fue adicionado. ¿Por qué no ordenó? Porque esta clase de aquí me devuelve una lista nueva.

[06:15] Entonces, aquí tenemos que utilizar la interface list String cursosList, por ejemplo, que va a ser el nombre de nuestra variable y vamos a importar este list también del paquete java.util. Ahora imprimimos cursosList. ¿Qué sucedió? Ahora ordenó de menor a mayor.

[06:54] Entonces, en esta clase 3 hemos visto 3 formas de ordenar: utilizando la clase collections, utilizando el método sort del mismo ArrayList o también utilizando streams, que más adelante vamos a ver más funciones y más métodos de cómo usar los streams para nuestras listas. Esto sería toda nuestra clase 3.